In the Claims:

1. (currently amended) A method of detecting changes in a continuous stream of channel associated signalling (CAS) data for a plurality of communication channels, the CAS data comprising successive blocks of data transmitted in series, each block of data comprising a plurality of timeslots of data transmitted in series, each timeslot of data comprising a plurality of CAS data bits for a communication channel, the method comprising the steps of:

(i) writing a block of data to an area of a circular memory buffer as a plurality of rows, each row comprising a predetermined number of timeslots of data;

(ii) writing a next block of data to an area of the circular memory buffer located sequentially after the area occupied by the previous block of data as a plurality of rows, each row comprising the predetermined number of timeslots of data, wherein after writing each row of said next block of data, changes in the data contained in the row are determined by comparing the row with the corresponding row in the previous block of data, wherein the comparing of a first row is commenced before a last row has been written; and

(iii) repeating step (ii) a plurality of times.

2. (original) The method of claim 1, wherein, in step (ii), comparing the row with the corresponding row in the previous block comprises:

locating the corresponding row in the previous block by applying a fixed memory offset from the location of the row;

reading the corresponding row in the previous block; and

comparing the CAS data bits in each timeslot of the row with the CAS data bits in each timeslot of the corresponding row of the previous block.

3. (original) The method of claim 1, wherein data is written to the circular memory buffer by direct memory access (DMA).

4. (original) The method of claim 1, wherein, in step (ii), after writing each row of the block of data, an interrupt is generated, and wherein changes in the data contained in the row are determined in response to the interrupt.

5. (original) The method of claim 1, wherein a row of data is written to the circular memory buffer every 125 μs.

6. (original) The method of claim 1, wherein all blocks of data are alternately written to one of two areas of the circular memory buffer.

7. (original) The method of claim 6, wherein the size of each areas of the circular memory buffer is equal to the size of a block of data.

8. (original) The method of claim 6, wherein the locations of the two areas of the circular memory buffer are consecutive.

9. (original) The method of claim 2, wherein the size of the fixed memory offset is equal to the size of a block of data.

10. (currently amended) A computer software product encoded to cause a computer to execute a method of detecting changes in a continuous stream of channel associated signalling (CAS) data for a plurality of communication channels, the CAS data comprising successive blocks of data transmitted in series, each block of data comprising a plurality of timeslots of data transmitted in series, each timeslot of data comprising a plurality of CAS data bits for a communication channel, the method comprising the steps of:

(i) writing a block of data to an area of a circular memory buffer memory as a plurality of rows, each row comprising a predetermined number of timeslots of data;

(ii) writing a next block of data to an area of the circular memory buffer located sequentially after the area occupied by the previous block of data as a plurality of rows, each row comprising the predetermined number of timeslots of data, wherein after writing each row of said next block of data, changes in the data contained in the row are determined by comparing the row with the corresponding row in the previous block of data, wherein the comparing of a first row is commenced before a last row has been written; and

(iii) repeating step (ii) a plurality of times.

11. (original) The computer software product of claim 10 arranged so that, in step (ii), comparing the row with the corresponding row in the previous block comprises:

locating the corresponding row in the previous block by applying a fixed memory offset from the location of the row;

reading the corresponding row in the previous block; and comparing the CAS data bits in each timeslot of the row with the CAS data bits in each timeslot of the corresponding row of the previous block.

12. (original) The computer software product of claim 10 arranged so that data is written to the circular memory buffer by direct memory access (DMA).

13. (original) The computer software product of claim 10 arranged so that, in step (ii), after writing each row of the block of data, an interrupt is generated, and wherein changes in the data contained in the row are determined in response to the interrupt.

14. (original) The computer software product of claim 10 arranged so that a row of data is written to the circular memory buffer every 125 μs.

15. (original) The computer software product of claim 10 arranged so that all blocks of data are alternately written to one of two areas of the circular memory buffer.

16. (original) The computer software product of claim 15 arranged so that the size of each area of the circular memory buffer is equal to the size of a block of data.

17. (original) The computer software product of claim 15 arranged so that the locations of the two areas of the circular memory buffer are consecutive.

18. (original) The computer software product of claim 11 arranged so that the size of the fixed memory offset is equal to the size of a block of data.

19. (currently amended) A processor and memory arrangement for use in detecting changes in a continuous stream of channel associated signalling (CAS) data for a plurality of communication channels, the CAS data comprising successive blocks of data transmitted in series, each block of data comprising a plurality of timeslots of data transmitted in series, each timeslot of data comprising a plurality of CAS data bits for a communication channel, the arrangement comprising:

a circular memory buffer; and

a processor arranged to:

(i) write a block of data to an area of the circular memory buffer as a plurality of rows, each row comprising a predetermined number of timeslots of data;

(ii) write a next block of data to an area of the circular memory buffer located sequentially after the area occupied by the previous block of data as a plurality of rows, each row comprising the predetermined number of timeslots of data, and, after writing each row of said next block of data, determine changes in the data contained in the row by comparing the row with the corresponding row in the previous block of data, wherein the comparing of a first row is commenced before a last row has been written; and

(iii) repeat step (ii) a plurality of times.

20. (original) The processor and memory arrangement of claim 19 arranged so that, in step (ii), comparing the row with the corresponding row in the previous block comprises:

locating the corresponding row in the previous block by applying a fixed memory offset from the location of the row;

reading the corresponding row in the previous block; and

comparing the CAS data bits in each timeslot of the row with the CAS data bits in each timeslot of the corresponding row of the previous block.

21. (original) The processor and memory arrangement of claim 19 arranged so that data is written to the circular memory buffer by direct memory access (DMA).

22. (original) The processor and memory arrangement of claim 19 arranged so that, in step (ii), after writing each row of the block of data, an interrupt is generated, and wherein changes in the data contained in the row are determined in response to the interrupt.

23. (original) The processor and memory arrangement of claim 19 arranged so that a row of data is written to the circular memory every 125 μs.

24. (original) The processor and memory arrangement of claim 19 arranged so that all blocks of data are alternately written to one of two areas of the circular memory.

25. (original) The processor and memory arrangement of claim 24, wherein the size of each areas of the circular memory is equal to the size of a block of data.

26. (original) The processor and memory arrangement of claim 24, wherein the locations of the two areas of the circular memory are consecutive.

27. (original) The processor and memory arrangement of claim 20 arranged so that the size of the fixed memory offset is equal to the size of a block of data.

28. (currently amended) A communication network node comprising:

a switch for routing trunks of communication channels; and

a processor and memory arrangement for use in detecting changes in a continuous stream of channel associated signalling (CAS) data for a plurality of communication channels, the CAS data comprising successive blocks of data transmitted in series, each block of data comprising a plurality of timeslots of data transmitted in series, each timeslot of data comprising a plurality of CAS data bits for a communication channel, the arrangement comprising:

a circular memory buffer; and

a processor arranged to:

(i) write a block of data to an area of the circular memory buffer as a plurality of rows, each row comprising a predetermined number of timeslots of data;

(ii) write a next block of data to an area of the circular memory buffer located sequentially after the area occupied by the previous block of data as a plurality of rows, each row comprising the predetermined number of timeslots of data, and, after writing each row of said next block of data, determine changes in the data contained in the row by comparing the row with the corresponding row in the previous block of data, wherein the comparing of a first row is commenced before a last row has been written; and

(iii) repeat step (ii) a plurality of times.